

7 SRv6

This topic introduces IPv6 Segment Routing (SRv6) and basic principles of SRv6 VPN, and provides an example for configuring an SRv6 VPN.

[7.1 Feature Change History](#)

[7.2 Introduction](#)

[7.3 Basic Principles](#)

[7.4 IS-IS for SRv6](#)

[7.5 SRv6 VPN](#)

[7.6 Example for Configuring of SRv6 VPN](#)

This section provides an example for configuring an SRv6 tunnel to carry IPv4 VPN services.

[7.7 Example for Configuring SRv6 EVPN](#)

7.1 Feature Change History

Product Version	Change Description
V100R023C10	Added: 7.7 Example for Configuring SRv6 EVPN
V100R019C1X and later versions	First version to support this feature

7.2 Introduction

Definition

Segment Routing IPv6 (SRv6) is a protocol designed to forward IPv6 data packets on a network based on source routes. IPv6 forwarding plane-based SRv6 enables

the ingress to add a segment routing header (SRH) into IPv6 packets. An explicit IPv6 address stack is pushed into the SRH. Transit nodes continue to update IPv6 destination IP addresses and offset the address stack to implement per-hop forwarding.

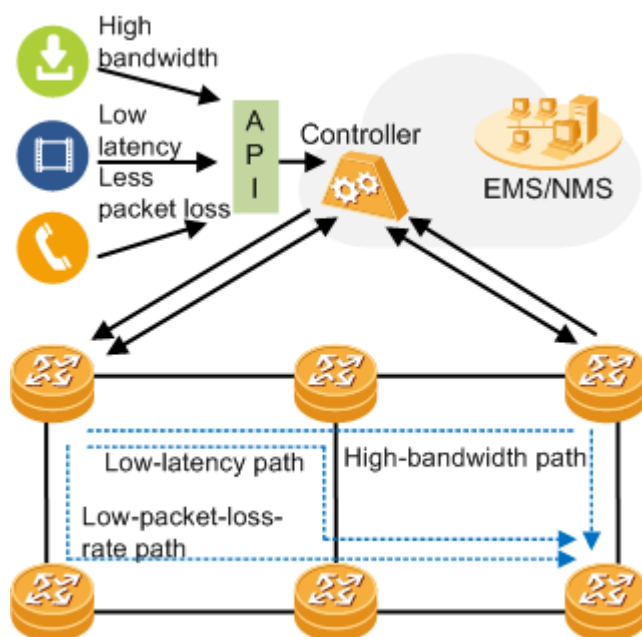
Purpose

Nowadays, different types of network services keep emerging, posing various requirements on the network. For example, real-time UC&C applications require low-latency and low-jitter paths, while big data applications require high-bandwidth paths with low packet loss rate. Traditional network adaptation methods are overstretched in meeting these new requirements, and make network deployment and O&M more and more difficult.

Simplified, low-latency, and SDN/NFV-based networks are the future network development trend. To meet the development requirements of networks, users require simpler VPN implementation using IPv6 addresses.

Segment Routing IPv6 (SRv6) came about in this context. [Figure 7-1](#) illustrates the network architecture defined by different services. SRv6 uses existing IPv6 forwarding technologies and extends IPv6 packet headers to implement label forwarding-like processing. Using SRv6, some IPv6 addresses are defined as instantiated segment IDs (SIDs). Each SID has its own explicit functions. SIDs are operated to flexibly plan paths and implement simplified VPNs.

Figure 7-1 Service-driven network



Benefits

SRv6 offers the following benefits to users:

- Streamlines network configurations to more easier to implement VPNs.

SRv6 does not use MPLS techniques and is fully compatible with existing IPv6 networks. Nodes merely support IPv6 forwarding, not MPLS forwarding.

Transit nodes can be incapable of SRv6 and forward IPv6 packets carrying the SRH over routes.

- Facilitates traffic optimization on IPv6 forwarding paths.
 SIDs with various service types are used to flexible plan explicit paths on the ingress to adjust service traffic.
- Better smooth evolution to SDN network.
 Segment Routing supports both traditional networks and SDN networks. It is compatible with existing equipment and ensures smooth evolution of existing networks to SDN networks instead of subverting existing networks.

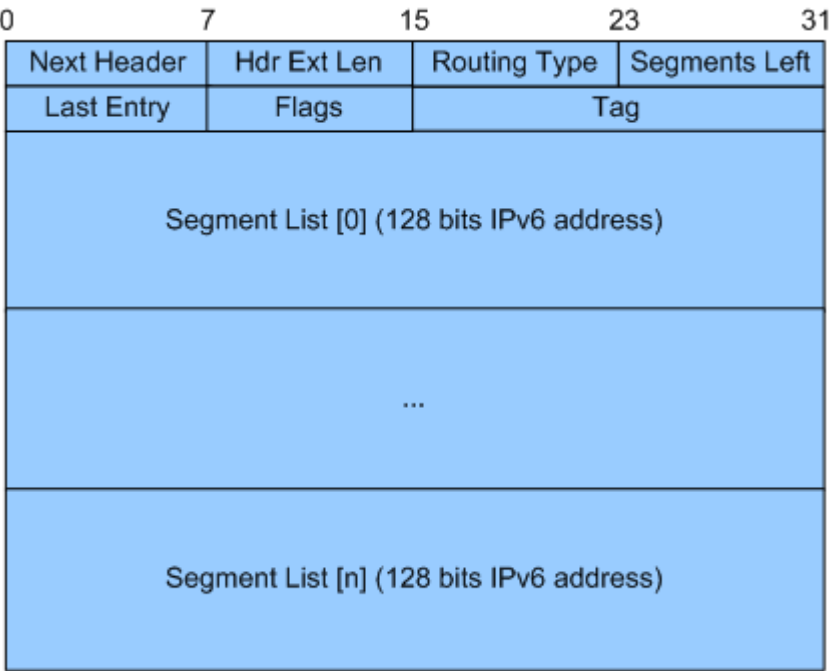
7.3 Basic Principles

SRH

An IPv6 packet consists of a standard IPv6 header, extended headers (0...n), and payload. To implement Segment Routing IPv6 (SRv6) based on the IPv6 forwarding plane, an IPv6 extension header, called segment routing header (SRH), is added. An SRH specifies an explicit path and stores IPv6 segment list information.

The ingress adds an SRH to an IPv6 packet, and each transit node forwards the packet based on path information carried in the SRH. [Figure 7-2](#) shows the SRH header format.

Figure 7-2 SRH format



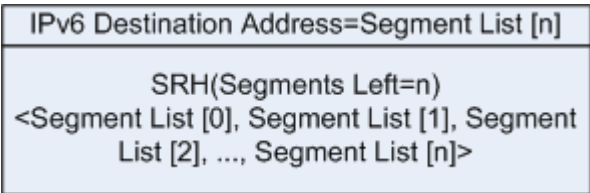
[Table 7-1](#) describes the fields carried in an SRH.

Table 7-1 Meanings of fields in an SRH

Field Name	Length	Description
Next Header	8 bits	Type of header immediately following the SRH.
Hdr Ext Len	8 bits	SRH header length. It covers the length from Segment List [0] to Segment List [n].
Routing Type	8 bits	Route header type. Value 4 identifies an SRH type.
Segments Left	8 bits	Number of transit nodes between the existing node and the egress.
Last Entry	8 bits	Index of the last segment in a segment list.
Flags	8 bits	Identifiers in a data packet.
Tag	16 bits	A tag indicating a packet as part of a class or group of packets.
Segment List[n]	128xn bits	Label segment list. A segment list is numbered from the last segment of a path. The Segment List is in the format of an IPv6 address.

Figure 7-3 shows the abstract SRH format.

Figure 7-3 Abstract SRH format



The meaning of each field is as follows:

- IPv6 Destination Address: IPv6 destination address in a packet. It is also called IPv6 DA. The IPv6 DA is a fixed value in an ordinary IPv6 packet. In SRv6, an IPv6 DA only identifies the next hop of an existing packet and is changeable.
- <Segment List [0], Segment List [1], Segment List [2],..., Segment List [n]>: SRv6 packet segment list. Similar to an MPLS label stack in SR MPLS, it is generated on the ingress. The Segment List is encoded starting from the last segment of the path. For example, the first element of the segment list (Segment List [0]) contains the last segment of the path, the second element contains the penultimate segment of the path and so on.

In SRv6, each time a packet passes through an SRv6 node, the Segments Left (SL) field value decreases by 1, and the IPv6 DA changes. Both the Segments Left and Segments List fields determine IPv6 DA information.

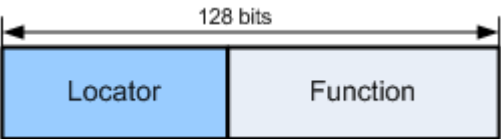
- If the SL value is n, the IPv6 DA value is equal to the Segments List [n] value.
- ...
- If the SL value is 1, the IPv6 DA value is equal to the Segments List [1] value.
- If the SL value is 0, the IPv6 DA value is equal to the Segments List [0] value.

SRv6 Segment

An SRv6 Segment is a 128-bit value, which is also called an SRv6 SID or SID. A SID is an instantiated IPv6 address that is assigned a unique function. An SRv6 SID identifies a node, a link, an L2VPN or L3VPN, or a service. The SRv6 SID can define any network function.

An SRv6 SID is a network instruction that consists of the locator and function. The locator is responsible for routing and must be unique within an SR domain. The function identifies any function on a device, such as a forwarding action or a service. The SRv6 SID structure facilitates network programming. [Figure 7-4](#) shows the SRv6 SID structure.

Figure 7-4 SRv6 SID



SRv6 SIDs are of various types. Each type of SRv6 SID indicates a specific function. End SID and End.X SID are the most common SRv6 SIDs. An End.DT4 SID stands for an IPv4 VPN. [Table 7-2](#) lists SRv6 segment types.

Table 7-2 SRv6 segment types

Label	Generat ion Mode	Function	Example
End SID	Manuall y configur ed.	Endpoint SID, which is used to identify the prefix of a destination address on a network. The End SID is similar to the Prefix Segment in SR MPLS. An IGP floods the End SID to the other NEs. The End SID is visible globally and takes effect globally.	Figure 7-5
End.X SID	Manuall y configur ed.	Endpoint SID of a Layer 3 cross-connection, which is used to identify a link on a network. The End.X SID is similar to Adjacency Segment in SR MPLS. An IGP floods the End.X SID to the other NEs. The End.X SID is visible globally and takes effect locally.	Figure 7-6

Label	Generat ion Mode	Function	Example
End.DT4 SID	Automa tically generat ed path name.	An End.DT4 SID stands for a PE endpoint SID that identifies an IPv4 VPN instance on a network. The forwarding behavior mapped to an End.DT4 SID is to decapsulate packets and searches the routing table of an IPv4 VPN instance for an entry to forward the packets. The End.DT4 SID is an equivalence to an IPv4 VPN label used in L3VPN scenarios.	Figure 7-7

Figure 7-5 End SID



Figure 7-6 End.X SID

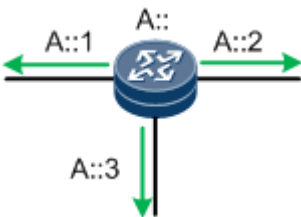
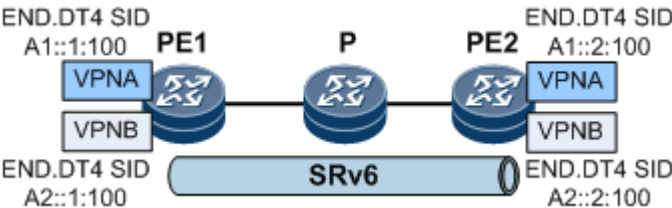


Figure 7-7 End.DT4 SID



An SRv6-capable node maintains a local SID table. This table contains all SRv6 segment information generated by the local node. Based on the table, the local node generates an SRv6 forwarding table (FIB). The local SID table provides the following functions:

- Defines a locally generated SID, for example, End.X SID.
- Specifies bindings to the SID.
- Stores related parameters.

Data Forwarding

SRv6 data is forwarded based on End SIDs and End.X SIDs. After receiving SRv6 packets, a node searches the local SID table based on IPv6 destination addresses (IPv6 DAs) and checks whether End or End.X SIDs are used.

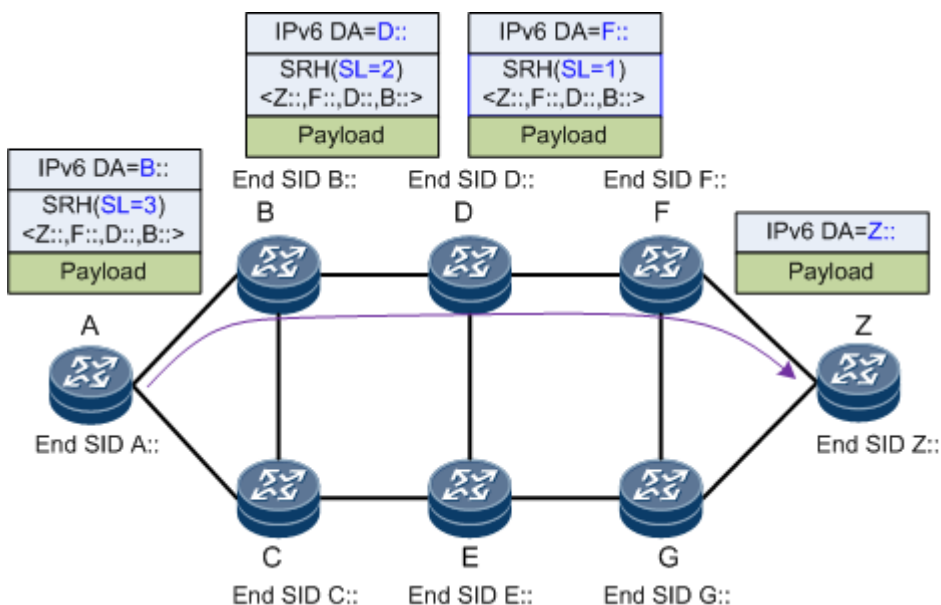
- If End SIDs are used, the node searches the IPv6 FIB table, finds a matching outbound interface and a next hop, and forwards the packets.
- If End.X DISs are used, the node forwards the packets through an outbound interface to a next hop specified in an End.X SID.

End SID-based data forwarding

Figure 7-8 demonstrates the process of forwarding data based on End SIDs:

1. Node A pushes an SRH into a data packet. The path information is <Z::, F::, D::, B::>, and the IPv6 DA is B::.
2. Each time the packet passes through a node, for example, nodes B and D, a node searches the local SID table based on the IPv6 DA, checks the End type, searches the IPv6 FIB table, finds a matching outgoing interface and next hop, reduces the SL value by one, and changes the IPv6 DA at a time.
3. After the packet arrives at node F, node F searches the local SID table based on the IPv6 DA, checks that the End type is used, continues to query the IPv6 FIB table, and finds the outgoing interface and next-hop IP address. In addition, node F reduces the SL value to 0 and changes the IPv6 DA to Z::. The path information <Z::, F::, D::, B::> becomes meaningless, and therefore, node F uses the PSP to remove SRH path information and forward the packet to node Z.

Figure 7-8 End SID-based data forwarding



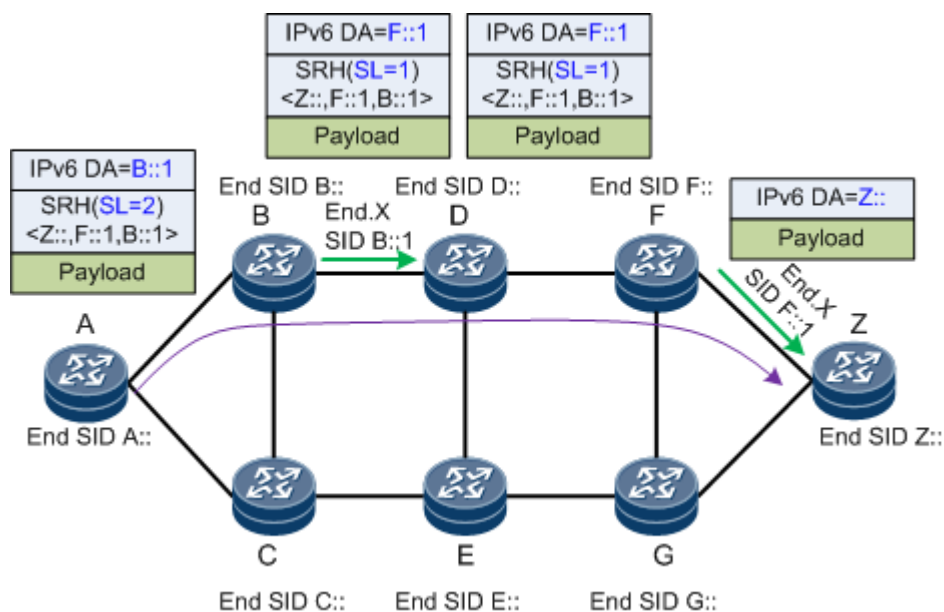
End.X SID-based data forwarding

Figure 7-9 demonstrates the process of forwarding data based on End.X SIDs:

1. Node A pushes an SRH into a data packet. The path information is <Z::, F::1, B::1>, and the IPv6 DA is B::1.
2. After the packet arrives at node B, node B searches the local SID table based on the IPv6 DA, checks that the End.X type is used, forwards the packet through the outgoing interface to the next hop, reduces the SL value by one, and changes the IPv6 DA to F::1.

- After the packet arrives at node D, node D searches the local SID table based on the IPv6 DA of F::1, finds no match, continues to use the IPv6 DA to query the FIB table, and finds a matching entry to forward the packet to node F.
- After the packet arrives at node F, node F searches the local SID table based on the IPv6 DA, checks that the End.X type is used, forwards the packet through the outgoing interface to the next hop, reduces the SL value to 0, and changes the IPv6 DA to Z::. The path information <Z::, F::1, B::1> becomes meaningless, and therefore, node F uses the PSP to remove SRH path information and forward the packet to node Z.

Figure 7-9 End.X SID-based data forwarding

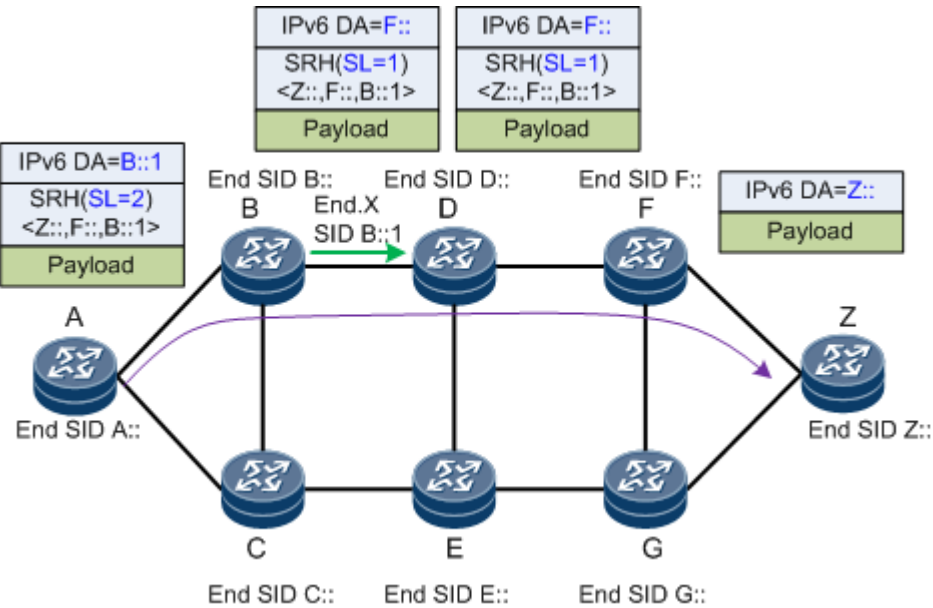


Data forwarding based on End SIDs and End.X SIDs

Figure 7-10 demonstrates the process of forwarding data based on End SIDs and End.X SIDs:

- Node A pushes an SRH into a data packet. The path information is <Z::, F::, B::1>, and the IPv6 DA is B::1.
- After the packet arrives at node B, node B searches the local SID table based on the IPv6 DA, checks that the End.X type is used, forwards the packet through the outgoing interface to the next hop, reduces the SL value by one, and changes the IPv6 DA to F::.
- After the packet arrives at node D, node D searches the local SID table based on the IPv6 DA of F::, finds no match, continues to use the IPv6 DA to query the FIB table, and finds a matching entry to forward the packet to node F.
- After the packet arrives at node F, node F searches the local SID table based on the IPv6 DA, checks that the End type is used, forwards the packet through the outgoing interface to the next hop, reduces the SL value to 0, and changes the IPv6 DA to Z::. The path information <Z::, F::, B::1> becomes meaningless, and therefore, node F uses the PSP to remove SRH path information and forward the packet to node Z.

Figure 7-10 Data forwarding based on End SIDs and End.X SIDs



SRH Information Popping

SRv6 supports the penultimate segment POP of the SRH (PSP) function, which is similar to Penultimate Hop Popping (PHP) in MPLS. PSP is enabled to allow the penultimate segment node to remove the SRH that becomes meaningless before a packet is sent to the egress, which reduces the burden on the egress. If PSP is disabled, the SRH is removed on the egress.

7.4 IS-IS for SRv6

Segment routing uses an IGP to advertise topology information, prefix information, a segment routing global block (SRGB), and label information. To complete the preceding functions, the IGP extends some TLVs of protocol packets. Table 7-3 describes TLVs of the IS-IS SRv6 extension.

Table 7-3 TLVs of the IS-IS SRv6 extension

Type	Name	Function	Position
TLV	SRv6 SID TLV	Used to advertise SRv6 SIDs.	IS-IS packets.
Sub-TLV	SRv6 Capabilities sub-TLV	Used to advertise SRv6 capabilities.	IS-IS TLV of type 242.
	P2P SRv6 X-SID sub-TLV	Used to advertise SRv6 SIDs on a P2P network.	IS-IS TLV of types 22, 23, 141, 222, 223, and so on.
	LAN SRv6 X-SID sub-TLV	Used to advertise SRv6 SIDs on a LAN.	IS-IS TLV of types 22, 23, 141, 222, 223, and so on.

Type	Name	Function	Position
Sub-sub-TLV	SRv6-function Descriptor	Used to advertise SRv6 SID function types.	SRv6 SID TLV, P2P SRv6 X-SID sub-TLV, and LAN SRv6 X-SID sub-TLV. In the SRv6 SID TLV, this field functions as a sub-TLV. In the P2P SRv6 X-SID sub-TLV and LAN SRv6 X-SID sub-TLV, this field functions as a sub-sub-TLV.

SRv6-Function Descriptor

The SRv6-function Descriptor defines the function type bound to each SRv6 SID. [Figure 7-11](#) shows the format of the SRv6-function Descriptor.

Figure 7-11 SRv6-function Descriptor Sub-sub-TLV

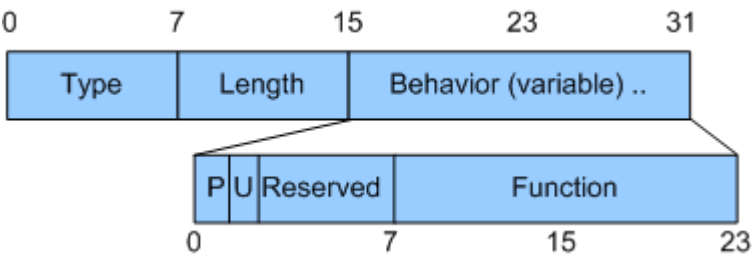


Table 7-4 Meanings of fields in the SRv6-function Descriptor Sub-sub-TLV

Field Name	Length	Description
Type	8 bits	Type.
Length	8 bits	Length. The value is three times the number of functions.

Field Name	Length	Description
Behavior	Variable length	Function flag: • P: PSP flag • U: USP flag • Reserved: Reserved bits • Function: function type: – 0: Reserved – 1: End – 2: End.DX6 – 3: End.DT6 – 4: End.X

7.5 SRv6 VPN

SRv6 VPN transmits VPN data along SRv6 tunnels. [Table 7-5](#) describes the comparison between the SRv6 VPN and BGP/MPLS IPv6 VPN.

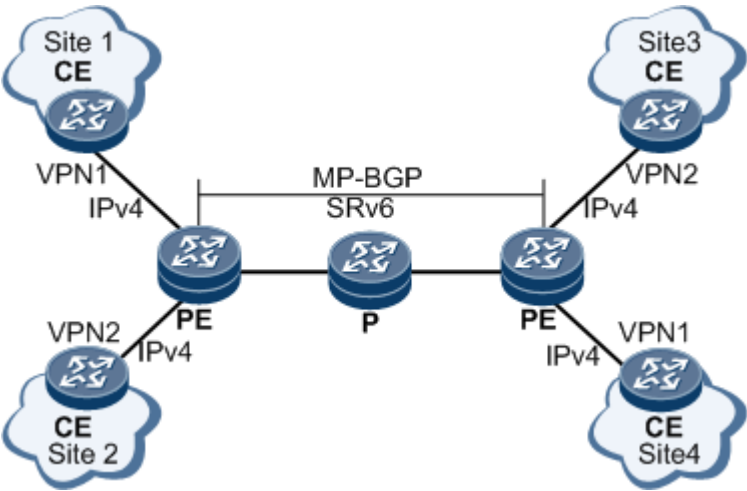
Table 7-5 Comparison between the SRv6 VPN and BGP/MPLS IPv6 VPN

Item	SRv6 VPN	BGP/MPLS IPv6 VPN
VPN service type	IPv4 VPN over SRv6 tunnel	IPv6 VPN over MPLS tunnel
VPNv4 route identifier and crossing	RD: identifies a specified VPN address space. RT: the local import RT must be the same as the peer export RT.	RD: identifies a specified VPN address space. RT: the local import RT must be the same as the peer export RT.
Route transfer	The IPv6 peer relationship is enabled in the BGP VPNv4 address family to transfer IPv4 route information.	The BGP VPNv4 address family is used to transfer IPv6 route information.
Public network route interworking	BGP + IS-IS	BGP + IGP or static route
VPN label	VPN labels do not exist. SRv6 VPN SIDs are used instead.	BGP assigns VPN labels. VPN labels are inner labels carried in BGP/MPLS IPv6 VPN public-network packets and used to identify VPN instances.

Item	SRv6 VPN	BGP/MPLS IPv6 VPN
MPLS label	MPLS labels do not exist.	MPLS LDP or TE assigns labels. MPLS labels are outer labels carried in BGP/MPLS IPv6 VPN public-network packets and used to identify tunnels.
Private network table lookup	The egress removes SRv6 VPN SIDs, identifies VPN instances based on SRv6 VPN SIDs, and searches the local SID table of each VPN instance.	The egress removes MPLS labels and VPN labels, identifies VPN instances based on VPN labels, searches VPN routing tables, and forwards packets over IP.

Figure 7-12 shows the typical SRv6 VPN networking.

Figure 7-12 Typical SRv6 VPN networking

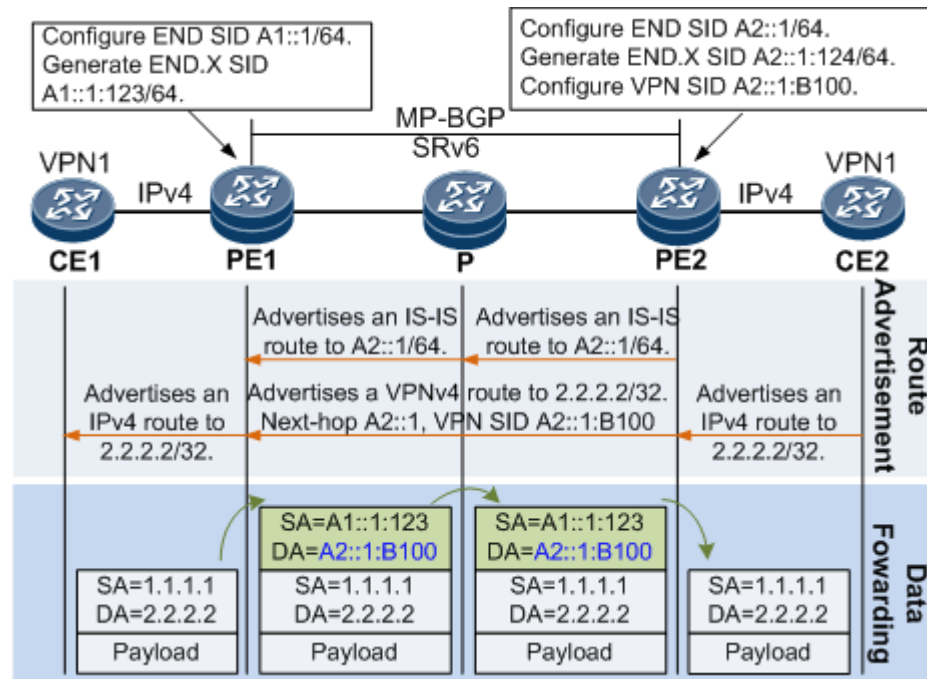


SRv6 VPN characteristics are as follows:

- Transmits packets using extended BGP.
- Encapsulates and transmits private network data packets over SRv6 LSPs or TE tunnels serving as public network tunnels.
- Allows a device that may play PE, P, and CE roles to play only a single role at a time.

IPv4 VPN over SRv6 allows SRv6 on public networks to carry IPv4 VPN data.

Figure 7-13 describes the process of advertising the routes and forwarding data in IPv4 VPN over SRv6.

Figure 7-13 Process of advertising the routes and forwarding data in IPv4 VPN over SRv6 LSP

In the route advertisement phase:

1. An END.SID is set on PE2 and advertised to PE1 using an IGP.
2. PE1 runs an IGP and learns the route (A2:1::/64) to the network mapped to the SRv6 SID.
3. PE2 automatically generates an END.DT4 SID (A2:1::B100) within the END.SID range for the VPN instance and a local SID table.
4. After receiving the VPNv4 route advertised by CE2, PE2 converts it to a BGP VPNv4 route and advertises the route to PE1 through a BGP peer relationship. The route carries the SRv6 VPN SID, which is END.DT4 SID (A2:1::B100) of the VPN instance.
5. Upon receipt of the VPNv4 route, PE1 injects the route to the VRF routing table, converts it to a common IPv4 route, and advertises it to CE1.

In the data forwarding phase:

1. CE1 sends a common IPv4 packet to PE1.
2. After receiving the private network packet on an interface bound to a VPN instance, PE1 searches a VRF routing table for an IPv4 prefix matching the destination and finds an associated SRv6 VPN SID and next hop information. PE1 encapsulates the SRv6 VPN SID (A2:1::B100) as a destination address into the IPv6 packet.
3. PE1 finds the route to A2:1/64 based on the longest match rule and forwards the packet to the P over the shortest path.
4. The P finds the route to A2:1/64 based on the longest match rule and forwards the packet to PE2 over the shortest path.
5. PE2 searches the local SID table for a behavior matching A2:1::B100, removes the IPv6 packet header, matches the packet against a VPN instance based on

the END.DT4 SID, and searches for a matching entry in the VRF routing table for forwarding. The packet is restored to a common IPv4 packet.

7.6 Example for Configuring of SRv6 VPN

This section provides an example for configuring an SRv6 tunnel to carry IPv4 VPN services.

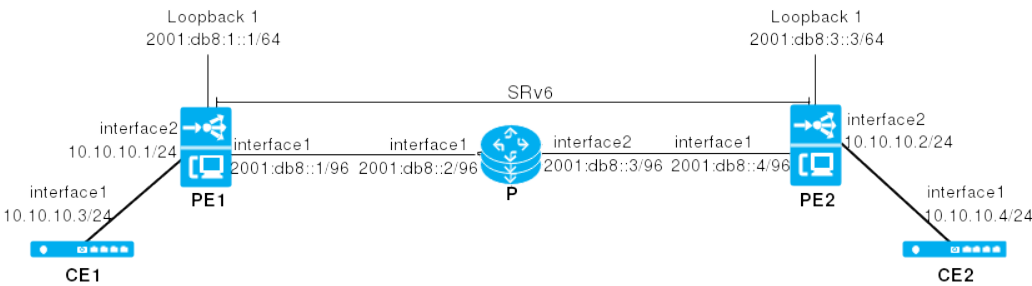
Service Requirements

IPv4 VPN over SRv6 indicates that an IPv6 tunnel on the public network is used to carry private network data of an IPv4 VPN.

As shown in [Figure1](#),

- PE1, P and PE2 belong to the same AS and run IS-IS to implement IPv6 network connectivity.
- PE1, P, and PE2 are Level-1 devices in area 1.

Figure 7-14 SRv6 VPN networking diagram



Configuration Flowchart

1. On each device, enable IS-IS, configure a level, and specify a network entity.
2. On PE1 and PE2, configure a VPN instance.
3. Between PE1 and the PE2, set up an IBGP peer relationship.
4. Between PE1 and PE2, configure an SRv6 tunnel for forwarding private network data.

Data preparation

To complete the configuration, you need the following data:

- IPv6 address and VLAN for each interface of PE1, P and the PE2.
- IPv4 address and VLAN for each interface of CE1 and CE2.
- Levels of PE1, P and PE2.
- AS number of PE1, P and the PE2.
- Name and RD of a VPN instance on PE1 and PE2.

[Table 7-6](#) provides the SRv6 VPN configuration data plan.

Table 7-6 SRv6 VPN configuration data plan

Item	Data
IS-IS parameters of PE1	IS-IS process ID: 1 Level: level-1 Network entity title: 10.0000.0000.0001.00
	IS-IS interface: Interface ID: 0/9/0 VLAN ID: 20 VLAN type: standard VLAN IPv6 address: 2001:db8::1/96
VPN parameters of PE1	Instance name: vrf100 RD: 100:1 Export VPN target: 1:1 Import VPN target: 1:1
	VPN interface: Upstream interface: 0/9/1 VLAN ID: 200 VLAN type: smart VLAN IPv4 address: 10.10.10.1/24
BGP parameter of PE1	BGP AS number: 100

Example

1. Configure the IS-IS function.

```
#Configure PE1.
huawei(config)#vlan 20 standard
huawei(config)#port vlan 20 0/9 0
huawei(config)#interface vlanif 20
huawei(config-if-vlanif20)#ipv6 enable
huawei(config-if-vlanif20)#ipv6 address 2001:db8::1 96
huawei(config)#interface loopback1
huawei(config-if-loopback1)#ipv6 enable
huawei(config-if-loopback1)#ipv6 address 2001:db8:1::1 64
huawei(config-if-loopback1)#quit
huawei(config)#isis 1
huawei(config-isis-1)#ipv6 enable
huawei(config-isis-1)#is-level level-1
huawei(config-isis-1)#network-entity 10.0000.0000.0001.00
huawei(config-isis-1)#quit
huawei(config)#interface vlanif 20
huawei(config-if-vlanif20)#isis ipv6 enable 1
huawei(config-if-vlanif20)#quit
huawei(config)#interface loopback1
huawei(config-if-loopback1)#isis ipv6 enable 1
huawei(config-if-loopback1)#quit
huawei(config)#isis 1
huawei(config-isis-1)#ipv6 enable
```

```
huawei(config-isis-1)#is-level level-1
huawei(config-isis-1)#network-entity 10.0000.0000.0001.00
huawei(config-isis-1)#quit
huawei(config)#interface vlanif 20
huawei(config-if-vlanif20)#isis ipv6 enable 1
huawei(config-if-vlanif20)#quit
huawei(config)#interface loopback1
huawei(config-if-loopback1)#isis ipv6 enable 1
huawei(config-if-loopback1)#quit
```

#Configure PE2. Configure the IS-IS function for PE2 in the same way as PE1.

#Configure P. Configure IS-IS for router P with reference to router configuration descriptions.

2. On a PE, configure a VPN instance for which the IPv4 address family is enabled, and connect a CE to a PE.

#Configure PE1.

```
huawei(config)#ip vpn-instance vrf100
huawei(config-vpn-instance-vrf100)#ipv4-family
huawei(config-vpn-instance-vrf100-af-ipv4)#route-distinguisher 100:1
huawei(config-vpn-instance-vrf100-af-ipv4)#vpn-target 1:1 export-extcommunity
huawei(config-vpn-instance-vrf100-af-ipv4)#vpn-target 1:1 import-extcommunity
huawei(config-vpn-instance-vrf100-af-ipv4)#quit
huawei(config)#vlan 200 smart
huawei(config)#port vlan 200 0/9 1
huawei(config)#interface vlanif 200
huawei(config-if-vlanif200)#ip binding vpn-instance vrf100
huawei(config-if-vlanif200)#ip address 10.10.10.1 24
huawei(config-if-vlanif200)#quit
```

#Configure PE2. Configure a Layer 3 VPN for PE2 in the same way as PE1.

#Configure CE1. Configure the IP address of interface 1 for CE1.

```
huawei(config)#vlan 200 smart
huawei(config)#interface vlanif 200
huawei(config-if-vlanif200)#ip address 10.10.10.3 24
huawei(config-if-vlanif200)#quit
```

#Configure CE2. Configure the IP address of interface 1 for CE2 in the same way as CE1.

Each PE can ping the connected CEs.

3. Between PE1 and the RR, set up an IBGP peer relationship.

#Configure PE1.

```
huawei(config)#bgp 100
huawei(config-bgp)#peer 2001:db8:3::3 as-number 100
huawei(config-bgp)#peer 2001:db8:3::3 connect-interface loopback 1
huawei(config-bgp)#ipv4-family vpnv4
huawei(config-bgp-af-vpnv4)#peer 2001:db8:3::3 enable
huawei(config-bgp-af-vpnv4)#peer 2001:db8:3::3 prefix-sid
huawei(config-bgp-af-vpnv4)#quit
huawei(config-bgp)#quit
```

#Configure PE2. Configure PE2 in the same way as PE1.

After the configuration is complete, run the display bgp vpnv4 all peer command on the PE. The command output shows that the EBGP peer relationship has been established between the PE and RR and the status of the EBGP peer relationship is established.

4. Between PEs, configure an SRv6 tunnel for forwarding private network data.

#Configure PE1.

```
huawei(config)#segment-routing ipv6
huawei(config-segment-routing-ipv6)#encapsulation source-address 2001:db8:11::11
huawei(config-segment-routing-ipv6)#locator test1 ipv6-prefix 1:: 64 static 32
```

```
huawei(config-segment-routing-ipv6-locator)#quit
huawei(config-segment-routing-ipv6)#quit
huawei(config)#bgp 100
huawei(config-bgp)#ipv4-family vpn-instance vrf100
huawei(config-bgp-vrf100)#segment-routing ipv6 best-effort
huawei(config-bgp-vrf100)#segment-routing ipv6 locator test1
huawei(config-bgp-vrf100)#quit
huawei(config-bgp)#quit
```

#Configure PE2. Configure PE2 in the same way as PE1.

You can run the **display segment-routing ipv6 locator verbose** command to query the Locator information of the IPv6 Segment Routing. And you can run the **display segment-routing ipv6 local-sid** command to query the Local SID information of the IPv6 Segment Routing.

5. Check the configuration result.
CEs in the same VPN can ping each other.

7.7 Example for Configuring SRv6 EVPN

This section provides an example for configuring a Segment Routing over IPv6 (SRv6) tunnel to carry Ethernet VPN (EVPN) services.

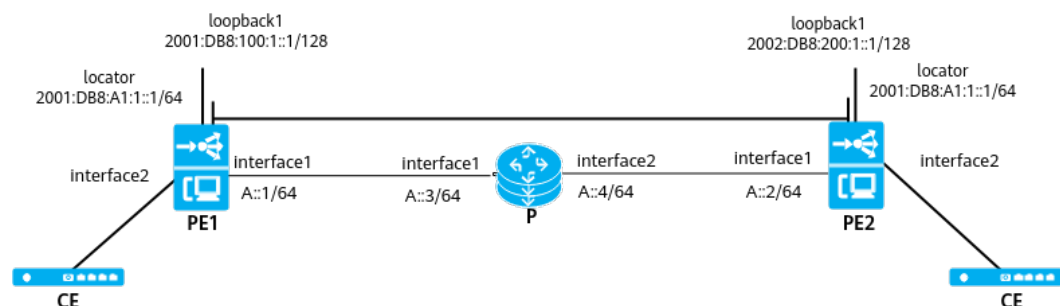
Service Requirement

The EVPN virtual private LAN service (VPLS) uses the EVPN Ethernet local area network (E-LAN) model to carry multipoint-to-multipoint (MP2MP) VPLSs. EVPN VPLS over SRv6 Best Effort (BE) builds on this technology and uses SRv6 BE paths over public networks to carry EVPN E-LAN services. Implementing EVPN VPLS over SRv6 BE involves establishing SRv6 BE paths, advertising EVPN routes, and forwarding data.

As shown in [Figure 7-15](#), PE1 and PE2 must communicate through an IPv6 public network and SRv6 BE is deployed on the public IPv6 network to carry EVPN E-LAN services.

- PE1, the P, and PE2 are in the same autonomous system (AS) and run the Intermediate System to Intermediate System (IS-IS) or Open Shortest Path First (OSPF) protocol to implement IPv6 network connectivity.
- In addition, a bidirectional SRv6 BE path is deployed between PE1 and PE2 to carry EVPN E-LAN services.

Figure 7-15 SRv6 EVPN networking diagram



Configuration Procedure

Configure SRv6 EVPN according to the following procedure:

1. Enable IS-IS, configure an IS-IS level, and specify a network entity title (NET) on PE1, the P, and PE2.
2. Configure an EVPN instance working in bridge domain (BD) mode on each PE and bind the corresponding BD to an access-side sub-interface.
3. Establish a BGP EVPN peer relationship between the PEs.
4. Configure SRv6 BE on the PEs.

Data Preparation

To complete the configuration, prepare the following data:

- IPv6 address of each interface on PE1, the P, and PE2 and VLAN planning
- Route distinguisher (RD) for EVPN instance evpn1
- BD ID: 10
- Locator name on PE1: a; dynamically generated opcodes
- Length of the field for locator a: 8 bits (*args*)

Example

1. Enable IS-IS, configure an IS-IS level, and specify an NET on PE1, the P, and PE2.

Configure PE1.

```
huawei(config)#vlan 20 standard
huawei(config)#port vlan 20 0/9 0
huawei(config)#service-port vlan 20 gpon 0/6/0 ont 0 gemport 1 multi-service user-vlan 20 tag-
transform translate
huawei(config)#vlan 10 standard
huawei(config)#port vlan 10 0/9 0
huawei(config)#interface vlanif 10
huawei(config-if-vlanif10)#ipv6 enable
huawei(config-if-vlanif10)#ipv6 address A::1/64
huawei(config-if-vlanif10)#ip address 10.10.10.1 255.255.255.0
huawei(config)#interface loopback1
huawei(config-if-loopback1)#ipv6 enable
huawei(config-if-loopback1)#ipv6 address 2001:DB8:100:1::1/128
huawei(config-if-loopback1)#quit
huawei(config)#isis 1
huawei(config-isis-1)#is-level level-1
huawei(config-isis-1)#cost-style wide
huawei(config-isis-1)#network-entity 10.0000.0000.0001.00
huawei(config-isis-1)#quit
huawei(config)#interface vlanif 10
huawei(config-if-vlanif10)#isis ipv6 enable 1
huawei(config-if-vlanif10)#quit
huawei(config)#interface loopback1
huawei(config-if-loopback1)#isis enable 1
huawei(config-if-loopback1)#isis ipv6 enable 1
huawei(config-if-loopback1)#quit
```

Configure PE2. Configure IS-IS on PE2 in the same way as on PE1.

Configure the P. Refer to the router configuration for configuring router P.

After the configuration is complete, run the **display isis peer** command to display the IS-IS neighbor information; run the **display isis route** command to display the routing table information.

2. Configure SRv6 BE between the PEs.

Configure PE1.

```
huawei(config)#segment-routing ipv6
huawei(config-segment-routing-ipv6)#encapsulation source-address 2001:DB8:100:1::1
huawei(config-segment-routing-ipv6)#locator a ipv6-prefix 2001:DB8:A1:1::1 64 args 8
huawei(config-segment-routing-ipv6-locator)#quit
huawei(config-segment-routing-ipv6)#quit
huawei(config)#isis 1
huawei(config-isis-1)#ipv6 enable topology ipv6
huawei(config-isis-1)#segment-routing ipv6 locator a
huawei(config-isis-1)#quit
huawei(config-evpn-instance-evpn1)#quit
```

Configure PE2. Configure PE2 in the same way as PE1.

3. Configure an EVPN instance working in BD mode on each PE, and bind the instance to the corresponding BD.

Configure PE1.

```
huawei(config)#evpn source-address 1.1.1.1
huawei(config)#evpn vpn-instance evpn1 bd-mode
huawei(config-evpn-instance-evpn1)#route-distinguisher 100:1
huawei(config-evpn-instance-evpn1)#segment-routing ipv6 best-effort
huawei(config-evpn-instance-evpn1)#segment-routing ipv6 locator a
huawei(config-evpn-instance-evpn1)#vpn-target 1:1 export-extcommunity
huawei(config-evpn-instance-evpn1)#vpn-target 1:1 import-extcommunity
huawei(config-evpn-instance-evpn1)#quit
huawei(config)#bridge-domain 10
huawei(config-bd-10)#evpn binding vpn-instance evpn1
huawei(config-bd-10)#evpn binding vlan 20
huawei(config-bd-10)#quit
```

Configure PE2. Configure PE2 in the same way as PE1.

4. Configure a BGP EVPN peer relationship between the PEs.

Configure PE1.

```
huawei(config)#bgp 100
huawei(config-bgp)#private-4-byte-as enable
huawei(config-bgp)#peer 2002:DB8:200:1::1 as-number 100
huawei(config-bgp)#peer 2002:DB8:200:1::1 connect-interface LoopBack1
huawei(config-bgp)#ipv4-family unicast
huawei(config-bgp-af-ipv4)#undo synchronization
huawei(config-bgp-af-ipv4)#quit
huawei(config-bgp)#l2vpn-family evpn
huawei(config-bgp-af-evpn)#policy vpn-target
huawei(config-bgp-af-evpn)#peer 2002:DB8:200:1::1 enable
huawei(config-bgp-af-evpn)#peer 2002:DB8:200:1::1 reflect-client
huawei(config-bgp-af-evpn)#peer 2002:DB8:200:1::1 advertise encap-type srv6
huawei(config-bgp-af-evpn)#quit
huawei(config-bgp)#quit
```

Configure PE2. Configure PE2 in the same way as PE1.

After the configuration is complete, run the **display bgp evpn peer** command on the PEs to check whether a BGP EVPN peer relationship has been established between them. The command output shows that the BGP EVPN peer relationship has been established between the PEs and is in the **Established** state.

5. Verify the configurations.

Run the **display segment-routing ipv6 local-sid { end-dt2u | end-dt2m } forwarding** command on each PE to check information about the local SID table of the SRv6 BE tunnel.

Run the **display bgp evpn all routing-table** command on each PE. The command output shows the EVPN route sent from the peer device.